Tackling Safe and Efficient Multi-Agent Reinforcement Learning via Dynamic Shielding (Student Abstract)

Wenli Xiao[1], **Yiwei Lyu**[2], **John M. Dolan**[3]

[1]School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China
[2]Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA
[3]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
wenlixiao@link.cuhk.edu.cn, {yiweilyu,jdolan}@andrew.cmu.edu

Abstract

Multi-Agent Reinforcement Learning (MARL) has been increasingly used in safety-critical applications but has no safety guarantees, especially during training. In this paper, we propose dynamic shielding, a novel decentralized MARL framework to ensure safety in both training and deployment phases. Our framework leverages Shield, a reactive system running in parallel with the reinforcement learning algorithm to monitor and correct agents' behavior. In our algorithm, shields dynamically split and merge according to the environment state in order to maintain decentralization and avoid conservative behaviors while enjoying formal safety guarantees. We demonstrate the effectiveness of MARL with dynamic shielding in the mobile navigation scenario.

Introduction

Multi-Agent Reinforcement Learning (MARL) is a promising approach to obtaining learning control policies for multi-agent decision-making tasks such as transportation management, motion control, and autonomous driving. However, applying MARL methods in safety-critical autonomous systems (e.g., autonomous driving cars) can cause havoc due to the lack of formal safety guarantees. In addition, traditional MARL approaches with behavior penalties (i.e., negative rewards for unsafe actions) cannot ensure safety in practice (ElSayed-Aly et al. 2021). Therefore, it is challenging to develop safe MARL systems that are provably trustworthy.

We consider Linear Temporal Logic (LTL) to express safety specifications. LTL offers the benefit of being able to express high-level safety specifications. For example, *agents should not collide with obstacles and other agents at timestep* $t \in \forall_t$ can be expressed as $\square\neg collision$ in LTL syntax. We consider translating the LTL safety specification into a safe language accepted by a deterministic finite automaton (DFA). In addition, we extend the definition of safe RL in (Alshiekh et al. 2018) to MARL in the following way:

Definition 1. *Safe MARL is the process of learning optimal policies for multiple agents while satisfying a LTL safety specification* ϕ^s *during the learning and execution phases.*

We focus on creating an efficient MARL algorithm that provides safety regarding definition 1. This paper presents the following **contributions**: 1) We propose a novel framework of *dynamic shielding*, a variant of traditional shielding that enables shields to adaptively split and merge to monitor MARL agents more efficiently. 2) We present a novel online shield synthesis algorithm that enables frequent refactoring of shields in real-time.

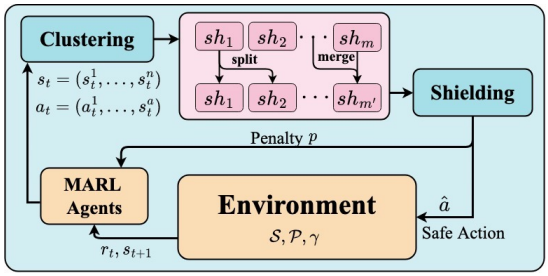

Figure 1: MARL with dynamic shielding framework.

Safe MARL with Dynamic Shielding

Our framework builds upon a method called Shield (Alshiekh et al. 2018), which stacks a layer between RL agents and the environment to monitor and correct agents' actions. When the RL agent attempts to take an unsafe action, the shield would correct it with a recoverable safe action and give a penalty. A Shield should have two properties: 1) Minimal interference. Namely, shields only correct an action if it violates the safety rule. 2) Correctness. Shields should detect every unsafe action and refine it with safe actions.

Our algorithm dynamically leverages shielding in the context of multi-agent scenarios. Dynamic shielding is a decentralized shield framework that synthesizes multiple shields to monitor agents concurrently, where each shield has two important operations: *merge* and *split*. Shields could merge into a larger shield to monitor a group of agents with shared state information. On the other hand, the computation complexity in shield synthesis increases along with the shield size. The split operation helps decrease computation costs when agents locate sparsely. Figure 1 shows that shields dynamically merge and split according to agents' states to achieve efficiency. There are three phases: 1) clustering, 2) shield reconstruction, and 3) shielding. In the clustering phase, the algorithm clusters agents into groups by their cur-

rent state. Then, in the shield re-construction phase, shields will merge or split to fit agents to the new group that formed in clustering. In the shielding phase, every shield will do shielding concurrently, which rejects agents' unsafe actions and replaces them with safety actions. Lastly, the MARL agents will be given an extra penalty for unsafe actions.

Synthesize Shields in Real-time

We **represent the shield** using a finite-state reactive system. According to the formulation in (ElSayed-Aly et al. 2021), a finite-state reactive system is a tuple $S = (Q, q_0, \Sigma_I, \Sigma_O, \delta, \lambda)$, where Σ_I and Σ_O are the I/O alphabets, Q is the state set, $q_0 \in Q$ denotes the initial state, $\delta : Q \times \Sigma_I \to Q$ is a transition function, and $\lambda : Q \times \Sigma_I \to \Sigma_O$ is an output function. Given the symbolic abstraction of the control input (i.e., input trace) $\overline{\sigma_I} = x_0 x_1 \ldots \in \Sigma_I^\infty$, the system S generates the trajectory of states (i.e., output trace) $\overline{\sigma_O} = \mathcal{S}(\overline{\sigma_I}) = \lambda(q_0, x_0)\lambda(q_1, x_1)\ldots \in \Sigma_O^\infty$, where $q_{i+1} = \delta(q_i, x_i)$ for all $i \geq 0$.

We **synthesize the shield** by solving a modified *two-player safety game*, a game played by the MARL agents and the environment, where the winning condition is defined by the LTL safety specification. We assume the state space has been converted into a symbolic abstraction given by a DFA $\mathcal{A}^e = (Q^e, q_0^e, \Sigma^e, \delta^e, F^e)$. We translate the LTL safety specification into another DFA $\mathcal{A}^S = \left(Q^S, q_0^S, \Sigma^S, \delta^S, F^S\right)$. We then combine $\mathcal{A}^e$ and $\mathcal{A}^S$ to formulate a game $\mathcal{G}^k = \left(G^k, g_0{}', \Sigma_1, \Sigma_2, \delta^{g\prime}, F^k\right)$, where the state space $G^k = G \times \{1...k\}$, the initial state $g_0{}' = (g_0, t = 1)$, the transition function $\delta^{g\prime}(g_t, t) = (\delta^g(g_t), t+1)$, and the winning condition $F^k = F \wedge (t \leq k)$. We can solve game $\mathcal{G}^k$ and compute the winning region $W \subseteq F^k$.We then construct the shield by translating $\mathcal{G}^k$ and W to a reactive system $S = (Q_\mathcal{S}, q_{0,\mathcal{S}}, \Sigma_{I,\mathcal{S}}, \Sigma_{O,\mathcal{S}}, \delta_\mathcal{S}, \lambda_\mathcal{S})$. Based on the nature of the formulated game, we can give the following provable safety guarantee:

Proposition 1. *Given a trace $s_0 a_0 s_1 a_1 \cdots \in (S \times A)^\omega$ jointly produced by MARL agents, the dynamic shielding, and the environment, state-action pair (s_t, a_t) is safe at every time step regarding definition 1.*

Experiments and Evaluations

Experiment Setup. We evaluate algorithms through navigation tasks in four different maps. Each map has four agents learning to navigate while avoiding obstacles in the environment. Each agent has the action space $\mathcal{A} = \{stay, up, down, left, right\}$. We set sparse goal-reaching rewards for this task: -1 for every valid step, a -10 collision penalty, and a $+100$ reward for arriving at the target.

Performance Evaluation. We integrate CQ-Learning (De Hauwere, Vrancx, and Nowé 2010) with the proposed dynamic shielding and factored shielding (ElSayed-Aly et al. 2021). We evaluate algorithms using the metrics such as maximum rewards, collision counts, and episode steps. Results in Figure 2 (upper) show that factored shielding (CQ+FS) and dynamic shielding (CQ+DS) can guarantee collision-free learning in all maps. Moreover, dynamic shielding obtains better policies with higher rewards

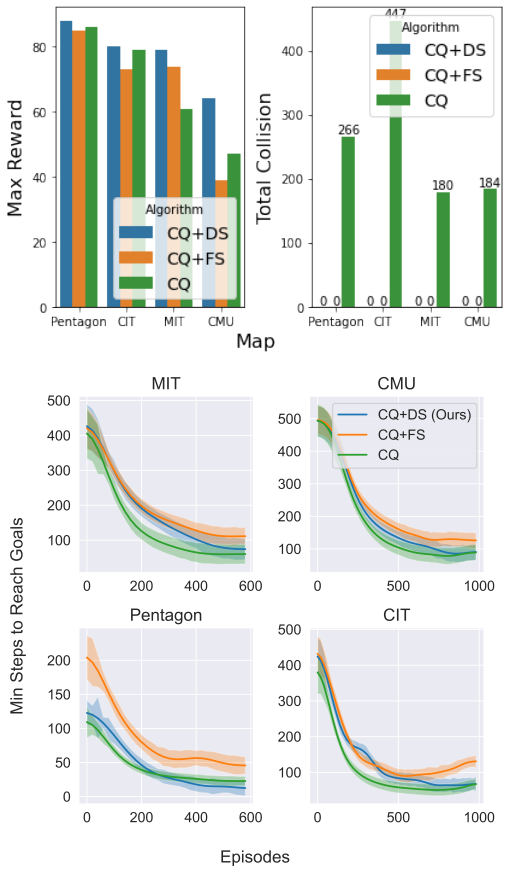

Figure 2: Comparing method performance.

compared to factored shielding and vanilla CQ. Figure 2 (lower) shows agents using proposed dynamic shielding need fewer steps to reach the target than factored shielding. Additionally, the dynamic shielding policy eventually has comparable performance to CQ without intervention, which we consider as the ground truth regarding the steps to reach the target. Therefore, the proposed dynamic shielding mitigates the conservative behaviors while ensuring safety.

References

Alshiekh, M.; Bloem, R.; Ehlers, R.; Könighofer, B.; Niekum, S.; and Topcu, U. 2018. Safe reinforcement learning via shielding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

De Hauwere, Y.-M.; Vrancx, P.; and Nowé, A. 2010. Learning multi-agent state space representations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, 715–722.

ElSayed-Aly, I.; Bharadwaj, S.; Amato, C.; Ehlers, R.; Topcu, U.; and Feng, L. 2021. Safe multi-agent reinforcement learning via shielding. *arXiv preprint arXiv:2101.11196*.